

Rapport de Drinking Machine

SI4

Yuqiao Zhao
Xin TANG

Rappel du but du projet :

Laisser structurer, déstructurer et restructurer un statechart décrivant le code de contrôle d'une machine à boisson. Plus précisément le but est de contrôler la fabrication et distribution de boissons selon les entrées d'un utilisateur. Plusieurs exigences sont données quand au processus possibles afin d'obtenir la boisson et sur les différentes actions possibles à chaque instant.

<u>Partie I : Les exigences implémentés</u>	<u>2</u>
<u>Partie II : Instruction de projet</u>	<u>2</u>
<u>a: Analyse de la structure</u>	<u>2</u>
<u>State—Choose</u>	<u>2</u>
<u>State-prepare</u>	<u>3</u>
<u>State after prepared</u>	<u>4</u>
<u>b. Explications spéciales:</u>	<u>4</u>
<u>Partie III : Validation & Verification</u>	<u>5</u>
<u>a. SELECT</u>	<u>5</u>
<u>b. PREPARE</u>	<u>6</u>
<u>c. MUTEX</u>	<u>6</u>
<u>Partie IV : Recul sur le travail</u>	<u>7</u>

Partie I : Les exigences implémentés

Expérimenté des efforts inlassables, on a implémenté tous le MVP(1-17) et tous le extensions(gestion de la soup, gestion de l'iced tea,gestion de l'avancement de la préparation,détection des gobelets).

Partie II : Instruction de projet

a: Analyse de la structure

1. State—Choose

Dans State-choose, nous avons un total de 8 états en parallèle pour la sélection de boissons.

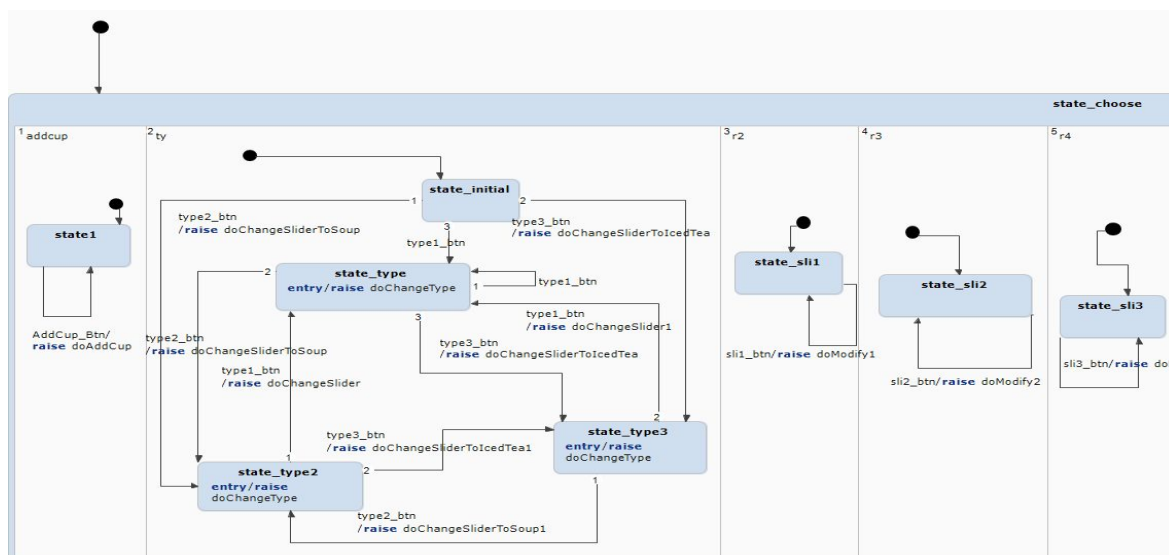
Ce qu'il faut expliquer, ce sont les deuxième, sixième, septième et huitième états.

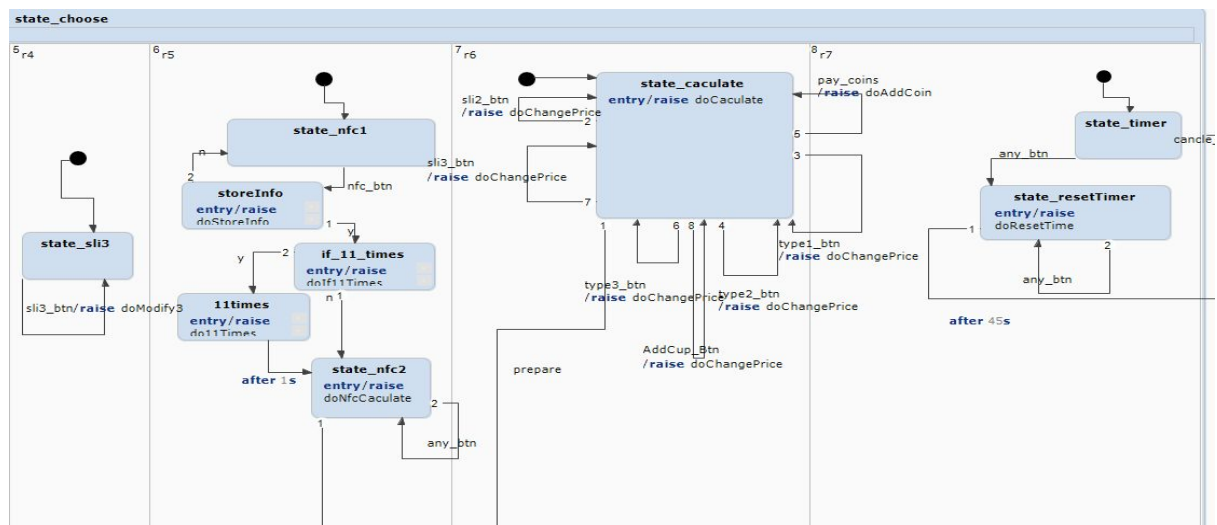
Le deuxième état est l'état composite, qui permet de réaliser le changement de Slider dû aux différentes boissons (soupe / thé glacé).

Le sixième état est l'état de conformité, qui est utilisé pour implémenter le stockage d'informations NFC. À cette fin, nous avons ajouté une boîte sur l'interface permettant aux utilisateurs de saisir les informations de leur carte bancaire.

Le septième état est l'état de calcul du prix actuel. Cet état intègre tous les facteurs qui peuvent modifier le prix actuel de la boisson et le montant total des pièces actuellement investies. La logique est que si le prix actuel payé est supérieur au prix d'entrée et que les ingrédients requis pour la boisson correspondante sont suffisants, la préparation sera appelée, sortira de l'état actuel et passera dans la deuxième étape - state-prepare.

Le dernier état est créé pour revenir à l'état initial sans opération pendant 45 s. Any_btn est un bouton virtuel, qui représente n'importe quel bouton et curseur de la machine. Lorsqu'un bouton ou un curseur est utilisé par l'utilisateur, il revient à plusieurs reprises à l'état 'resetTime'. Par conséquent, nous pouvons surveiller 45s sans opération via l'état 'resetTime', mettre fin à la commande, déterminer s'il est nécessaire de retourner la pièce, puis revenir à l'état initial.

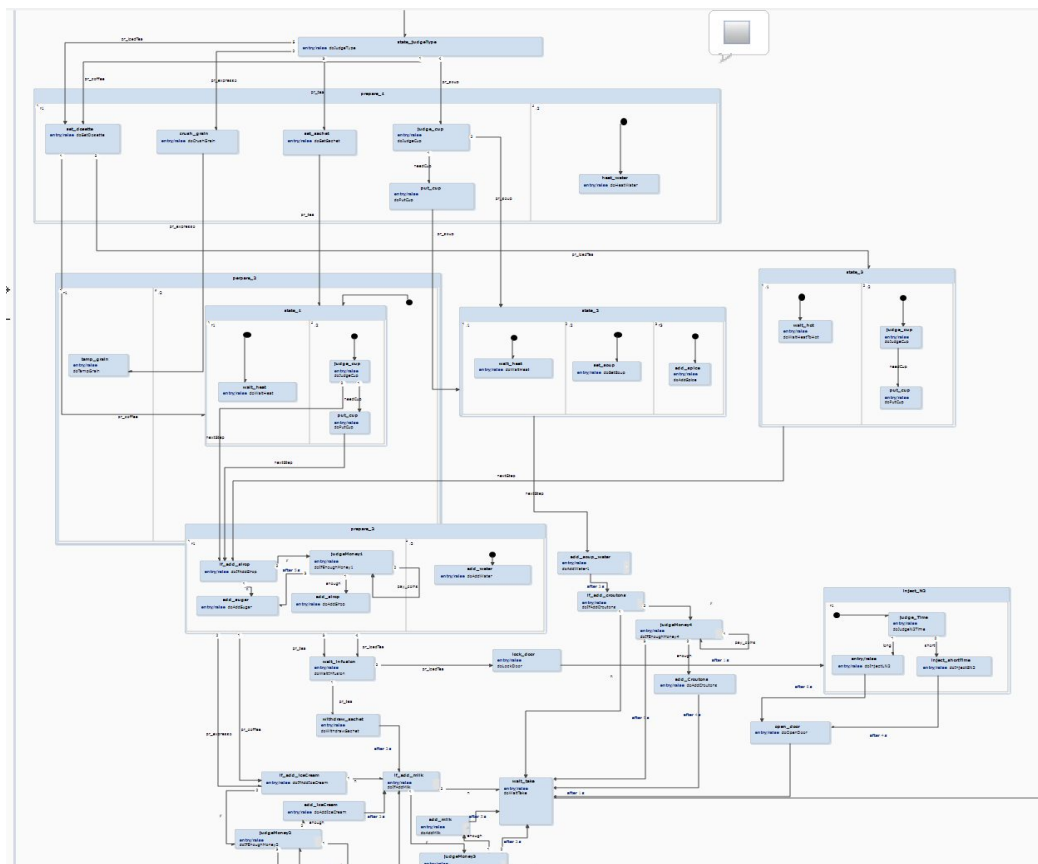




Cette partie de la fonction peut être réalisée en utilisant une structure parallèle ainsi qu'une structure séquentielle. Mais la structure parallèle est évidemment plus simple que la structure séquentielle. La raison en est que la sélection de la boisson (taille / goût sucré / température, etc.), la pièce de monnaie / carte et d'autres opérations ne sont pas dans l'ordre et peuvent être effectuées à volonté. Si une structure séquentielle est adoptée, le diagramme d'état deviendra un va-et-vient entre tous les deux états, ce qui est compliqué.

2. State-prepare

Au stade de la préparation des boissons, nous avons adopté une structure globale en couches et mis en parallèle dans chaque couche pour la réalisation en fonction des exigences du projet.

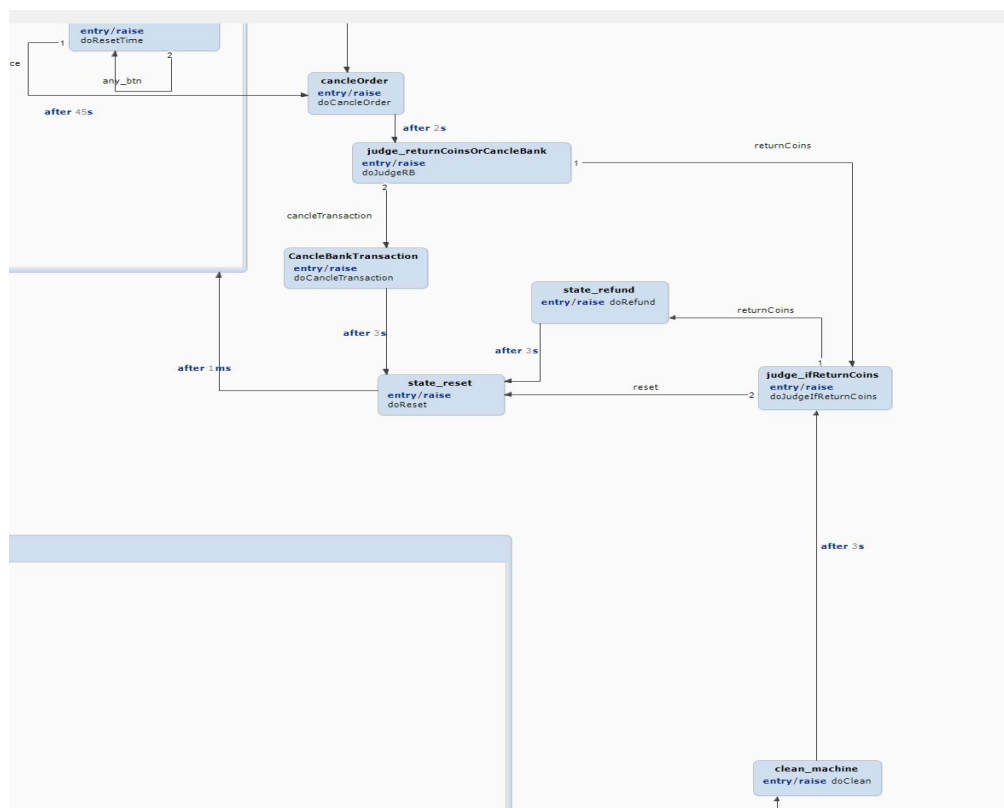


La raison de la structure en couches est que la séquence spécifique de préparation de la boisson ne peut pas être modifiée, et pour chaque couche en parallèle, c'est parce que plusieurs étapes de préparation sont effectuées au même stade en même temps.

En ce qui concerne le réglage de la période de temps de chaque étape de la partie de préparation, il convient de noter que s'il s'agit d'une pièce spéciale, telle que la récupération du sachet de thé, le temps de congélation de l'heure du thé glacé, etc., nous définissons directement l'heure spécifique dans le statut (4s / 8s etc.).

Pour les couches avec des parties communes, comme le chauffage de l'eau et l'attente de chauffage, nous n'avons pas clairement indiqué le temps spécifique requis pour l'achèvement de la couche, mais avons défini des temps différents dans le code à travers différentes boissons.

3. State after prepared



Dans la troisième partie—après avoir emporté la boisson, la structure que nous avons adoptée est une structure de jugement logique et de séquence relativement simple. Nous plaçons le nettoyage de la machine à boissons entre les deux états de retrait de la boisson et de détermination du remboursement, et définissons un état de réinitialisation d'état pour revenir à l'état initial, de sorte que la machine à boisson puisse fonctionner normalement.

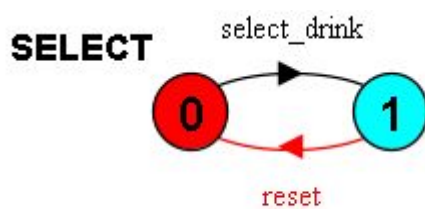
b. Explications spéciales:

1. Nous avons strictement limité la saisie des numéros de carte NFC. Les types non numériques ne peuvent pas être saisis et le numéro de carte bancaire analogique est composé de 16 chiffres. Si le numéro de carte bancaire saisi ne comporte pas 16 chiffres, il affichera Veuillez saisir le numéro de carte bancaire correct.

2. Pour les choix supplémentaires de l'utilisateur (options), nous avons mis en place deux boutons «oui» et «non» pour recevoir les choix de l'utilisateur.
3. Pour juger si les matières premières sont satisfaites avant de préparer la boisson, nous avons effectué un jugement numérique sur les quatre boissons restantes, à l'exception de la soupe, et défini le nombre de fois qu'elles peuvent être préparées à 100 fois. Si le nombre de fois dépasse 100, les matières premières sont insuffisantes. Pour la soupe, nous avons mis trois types d'assaisonnements (salt/basil/onion) pour cela, et mis chaque assaisonnement 100 fois.
4. Pour le prix de la boisson, nous fixons le prix requis par le projet lorsque la boisson est une petite tasse et que diverses options supplémentaires ne sont pas sélectionnées par défaut. Si la taille est moyenne, le prix de la boisson augmentera de 0,1 et le prix de la boisson augmentera de 0,3.

Partie III : Validation & Verification

a. SELECT



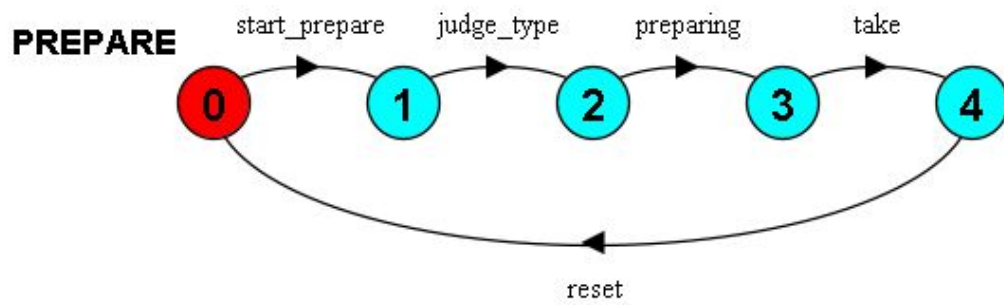
Verification :

Pour le module de sélection de boisson, chaque fois que nous sélectionnons, le statut de la boisson correspondante changera.

Validation:

En raison de la demande de l'utilisateur, le type de boisson peut être modifié selon les préférences de l'utilisateur.

b. PREPARE



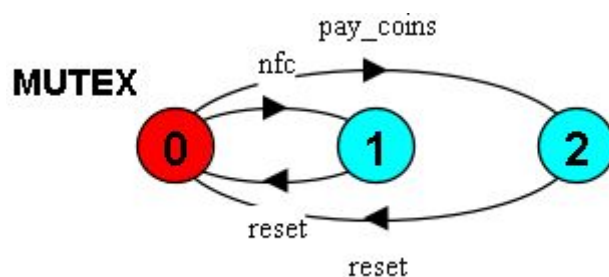
Verification :

Pour le processus de préparation de la boisson, en fonction du type de boisson, choisissez un processus de préparation différent, puis attendez que le client emporte la boisson.

Validation:

Puisque l'utilisateur a sélectionné la boisson, l'utilisateur n'est pas autorisé à poursuivre d'autres opérations pendant le processus de préparation correspondant.

c. MUTEX



Verification :

Le paiement par carte et le paiement en espèces sont des opérations mutuellement exclusives. Une fois le paiement par carte sélectionné, le paiement en espèces ne peut pas être sélectionné, et vice versa.



Validation:

Pour l'utilisateur, il n'est pas correct de payer en espèces et de glisser la carte en même temps.

Partie IV : Recul sur le travail

On pense que notre conception du processus de prise de la boisson par l'utilisateur n'est pas assez sophistiquée. On pense que nous pouvons déterminer si la boisson a été emportée en cliquant sur l'image de la boisson à l'écran. Etant donné que cela peut déterminer plus précisément si la boisson a été retirée, un temps d'avertissement peut être réglé juste après le processus d'attente, et la boisson sera invitée si elle n'a pas été retirée dans un laps de temps.

Pour le diagramme d'états, on a encore quelques domaines qui peuvent être améliorés. Par exemple, pour de nombreux états parallèles, on peut choisir un état de sortie unifié

 Exit Node , et pour certains jugements de sélection, on peut directement utiliser l'état de sélection  Choice .